#### [Consume PUT service in Angular and RESTful DELETE Service](javascript:void(0);)

#### **Objectives defined in the previous session is applicable for this session as well.** NOTE: There is no Quiz for this session

**Consume PUT Employee service in Angular component**   
  
Perform steps below to implement consumption of service to update employee details.  
  
**employee.ts**

* Modify the dateOfBirth data type as string, as we have included @JsonFormat which accepts date of birth as string.
* Update the changes in all the places where dateOfBirth value is defined in 'dd/mm/yyyy' format.

**employee.service.ts**

* Include constant for setting the content type of the request as JSON type. Include below code above @Injectable decorator defintion.

const httpOptions = {

  headers: new HttpHeaders({

    'Content-Type': 'application/json',

  })

};

* Include a new method updateEmployee which makes the PUT method call:

  updateEmployee(employee: Employee): Observable<void> {

    return this.httpClient.put<void>(this.employeeApiUrl, employee, httpOptions);

  }

**edit-employee.template.component.css**

* Include a new style for displaying success message

.success {

  color:green;

}

**edit-employee.template.component.ts**

* Include a new property error of type string for displaying the error message if the form submission returns error.
* Incorporate the below code in the onSubmit() event that is triggered when save button is clicked. The error section gets the message from the error response and sets the error property value.

  onSubmit() {

    this.employeeService.updateEmployee(this.employee).subscribe(

      data => {

        console.log('Employee update successful.');

        this.error = '';

      },

      error => {

        console.log(error);

        this.error = error.error.message;

        if (error.error.errors != null) {

          this.error = error.error.errors[0];

        }

      });

  }

**edit-employee.template.component.html**

* Include the below div above the Save button to display the error message.

    <div \*ngIf="error == ''" class="success">Employee details saved successfully.</div>

    <div \*ngIf="error != ''" class="text-error">{{error}}</div>

Test various scenarios as specified below by changing test date:

* Define dateOfBIrth value in incorrect date format
* Negative value for salary
* Success scenario where all field values are correct

**Consume PUT Employee service in reactive form component**   
  
Incorporate edit employee form in the edit employee reactive component.  
  
Consider below aspects when implementing:

* Reuse the updateEmployee() service method earlier
* On submission of Save button, get the employee form data from FormGroup object
* Refer previous hands on for rest of the implementation.

**Implement REST DELETE Service**   
  
Implement steps below to complete this:

* Implement a new delete service for employee by incorporating relevant code in EmployeeController, EmployeeService and EmployeeDao.
* The EmployeeDao should have the code to remove the respective item from the list or throw EmployeeNotFoundException if the id not found.
* Test the service and check if deletion happens correctly
* In employee-list component include a delete icon for each employee in the list
* On click of this icon include necessary coding in the component to invoke a new service that calls the new delete service created in the earlier steps.